

1.2.3.2 if 语句、switch 语句、多层条件语句

1. if 语句

if 语句分为单分支和双分支两种形式。

(1) 单分支，即如果表达式的值为真，则其后的语句块被执行；否则，其后的语句块略过。

```
int a,b,max;
scanf("%d %d",&a,&b); //读入两个数值,分别赋给 a 和 b
max = b; // 假设 b 最大
if (a > b)
    max = a; // 如果 a>b,那么更改 max 的值
printf("%d 和 %d 的较大值是: %d",a,b,max);
```

(2) 双分支，即如果表达式的值为真，则其后的语句块被执行；否则，执行 else 后的语句块。

```
int a,b,max;
scanf("%d %d",&a,&b);
if (a > b)
    max = a;
else
    max = b;
printf("%d 和 %d 的较大值是: %d\n",a,b,max);
```

if-else 语句也可以同时使用多个，构成多个分支，语法如下。

```
if (表达式 1) {
    语句块 1
} else if (表达式 2) {
    语句块 2
} else if (表达式 3) {
    语句块 3
}
...
else if (表达式 n) {
    语句块 n
} else {
    语句块 n + 1
}
```

2. switch 语句

switch 语句适用于表达数量较多的分支条件，语法如下。

```
switch ( 表达式 ) {  
case 常量表达式 1 : 语句块 1  
case 常量表达式 2 : 语句块 2  
...  
default : 语句块 n  
}
```

switch 语句可以包含多个 case 条件，case 后面的常量表达式必须各不相同。进入 switch 语句后，先计算表达式的值，然后逐一匹配每个 case 的常量表达式，当匹配成功时，执行对应的语句块，然后顺序执行之后的所有语句，直到整个 switch 语句结束。如果遇到一个 break 语句，则中止 switch 语句的执行。如果表达式与所有的 case 都不能成功匹配，则执行 default 后面的语句块，直到 switch 语句结束。

break 是专门用于跳出类似 switch 的分支和其他循环结构的语句。“跳出”是指一旦遇到 break，就不再执行 switch 中的其他语句，包括当前分支中的语句和其他分支中的语句。

default 负责处理不符合任何 case 的情况，而且 switch 语句中只能有一个 default 分支，它可以出现在开头或者结尾，也可以出现在两个 case 的中间，但 switch 语句中不是必须出现 default 选项的。

3. 多层条件语句

条件语句可以层层嵌套，形成多层条件语句。

参考词条

三目运算

延伸阅读

[1] KERNIHAN B W, RITCHIE D M. C 程序设计语言[M]. 徐宝文, 李志, 译. 2 版. 北京: 机械工业出版社, 2004: 45-48.

[2] BJARNE S. C++程序设计语言(第 1~3 部分)[M]. 王刚, 杨巨峰, 译. 4 版. 北京: 机械工业出版社, 2016: 196-200.

(金靖)

1.2.3.3 for 语句、while 语句、do while 语句

1. for 语句

for 语句可以控制一个语句或语句组重复执行指定的次数，语句块可以执行 0 或多次，直到给定的条件不成立。一般来说，在 for 语句开始时设定初始条件，比如给某个变量设定初始值。然后在 for 语句的每次循环中改变此变量的值，并将此变量作为判断循环是否结束的条件。

C++14 中有两种常见的使用 for 语句的方法：范围 for 和通用 for。

(1) 范围 for 可用于依次访问指定范围内的每个元素。

```
for (auto x : v) //v 可以是数组、vector、set 等
```

关键词 auto 的作用是自动类型推断，即根据 v 的值来推测变量 x 的类型。冒号之后的表达式必须是一个序列，变量 x 指向当前正在处理的元素，它类似于通用 for 中的 $a[i]$ ($a[]$ 为循环中调用的数组)。

(2) 通用 for 的用法如下所示。

```
for (表达式 1; 表达式 2; 表达式 3)
    语句块
```

通用 for 语句的执行顺序如下：

- ① 表达式 1 仅被执行 1 次，一般用作循环变量的初始化；
 - ② 表达式 2 作为循环是否继续的判断条件，如果条件为真则执行循环体中的语句块，否则循环结束；
 - ③ 表达式 3 在语句块执行后执行，一般为改变循环变量的值。
- 3 个表达式均可省略，此时 `for(;;)` 代表无限循环，只有遇到语句块中的 `break` 或 `return` 语句时才会退出循环。

2. while 语句

类似 for，while 也是实现循环结构的语句，用于重复执行循环体内的语句块，直到条件表达式的值为假，语法如下。

```
while (表达式)
    语句块
```

如果 while 括号内条件表达式的值为真，则执行语句块，之后程序控制转向，继续判断条件表达式是否成立。如果表达式的值为假，则不执行语句块，程序直接转到 while 语句后面的语句继续执行。

相比 for，while 语句更适合处理无明显循环变量的情况。while 也可以通过 `break` 或 `return` 语句结束。

3. do while 语句

do while 与 while 类似，用于重复执行循环体内的语句块，直到条件表达式的值为假，语法如下。

```
do
    语句块
while (表达式);
```

每一次循环中，先执行 do 语句块，如果 while 括号内条件表达式的值为真，则继续

执行 do 语句块。如果表达式的值为假，则不执行语句块，程序直接转到 do while 语句后面的语句继续执行。

do while 与 while 语句的区别在于表达式是在语句或语句组被执行之后计算的，所以 do 后面的语句块至少被执行 1 次。do while 语句同样可以通过 break 或 return 语句结束。

延伸阅读

[1] KERNIHAN B W, RITCHIE D M. C 程序设计语言[M]. 徐宝文, 李志, 译. 2 版. 北京: 机械工业出版社, 2004: 49-53.

[2] BJARNE S. C++程序设计语言(第 1~3 部分)[M]. 王刚, 杨巨峰, 译. 4 版. 北京: 机械工业出版社, 2016: 201-204.

(金靖)

1.2.3.4 多层循环语句

循环语句中的循环体又出现循环语句，则形成多层循环。

可用 break 强制结束循环，直接跳出该层循环(不会直接跳出所有循环)，也可用 continue 跳过当前循环中的代码，开始下一次循环。

在多层循环中，可使用 for、while、do while 等循环体，这些语句均可实现多层循环。需注意，每一层的循环变量互不相同，不得重复。

延伸阅读

KERNIHAN B W, RITCHIE D M. C 程序设计语言[M]. 徐宝文, 李志, 译. 2 版. 北京: 机械工业出版社, 2004: 51.

(金靖)

1.2.4 基本运算

1.2.4.1 算术运算：加、减、乘、除、整除、求余

算术运算符用于数值运算，包括加(+)、减(-)、乘(*)、除(/)、求余(%)等。

除法运算符在使用的时候需要注意，如果 a 、 b 是两个整数类型的表达式，那么 a/b 表示整除，值为 a 除以 b 的商的整数部分。只有 a 、 b 至少有一个为实型时， a/b 的值才是实型。

求余运算符“%”也称为取模运算符。它是双目运算符，两个操作数都是整数类型的， $a\%b$ 的值就是 a 除以 b 的余数。需注意，求余数的值的正负与 a 保持一致， b 在运算中取其绝对值。

参考词条

1. 高精度的加法

2. 高精度的减法
3. 高精度的乘法
4. 高精度整数除以单精度整数的商和余数
5. 成员函数和运算符重载

延伸阅读

KERNIHAN B W, RITCHIE D M. C 程序设计语言[M]. 徐宝文, 李志, 译. 2 版. 北京: 机械工业出版社, 2004: 32.

(金靖)

1.2.4.2 关系运算：大于、大于等于、小于、小于等于、等于、不等于

关系运算符用于表达式之间的大小比较，共有 6 种：大于(>)、小于(<)、等于(=)、大于等于(>=)、小于等于(<=)和不等($!=$)，都是双目运算符。

关系运算符的优先级低于算术运算符，高于赋值运算符。在六个关系运算符中，<、$=$、>、>=的优先级相同，高于==和!=，==和!=的优先级相同。

关系运算符运算的结果是布尔型，值只有两种：真或假，在表达式中，分别用 1 和 0 来代表。

延伸阅读

KERNIHAN B W, RITCHIE D M. C 程序设计语言[M]. 徐宝文, 李志, 译. 2 版. 北京: 机械工业出版社, 2004: 32-33.

(金靖)

1.2.4.3 逻辑运算：与(&&)、或(||)、非(!)

逻辑运算符，即对表达式或者变量做出逻辑判断的运算符，常用的有以下 3 种。

1. 与(&&)

双目运算符，&& 左右两边的值均为真时，结果才为真，否则为假。

2. 或(||)

双目运算符，|| 左右两边的值均为假时，结果才为假，否则为真。

3. 非(!)

单目运算符，如果表达式的值为真，那么“!表达式”的值为假，否则为真。

延伸阅读

KERNIHAN B W, RITCHIE D M. C 程序设计语言[M]. 徐宝文, 李志, 译. 2 版. 北京: 机械工业出版社, 2004: 32.

(金靖)

1.2.4.4 变量自增与自减运算

自增运算符“++”用于将整型或浮点型变量的值加 1，自减运算符“--”则是减